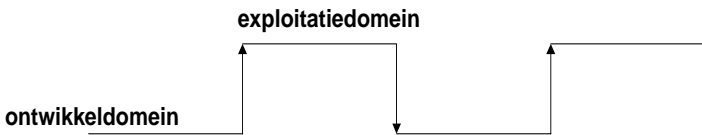


Onderwerp: <b>Configuration Management</b>		Onderdeel van: <b>Management Processen</b>		Versie: 0.9, 19 november 2002	
<b>DOEL</b> Het garanderen van de integriteit en consistentie van delen binnen een product, met andere woorden: <ul style="list-style-type: none"><li>• Voorkomen van tijdverlies door incompatibiliteit van deelproducten</li><li>• Identificeren, beheren en kunnen ‘terughalen’ van productversies (weten wat gewijzigd is en waarom)</li><li>• Het efficiënt kunnen reproduceren van problemen (en daardoor sneller kunnen oplossen)</li><li>• Mogelijk maken dat meerdere mensen gelijktijdig aan hetzelfde product werken</li></ul>					
 <p>Configuratie management in twee domeinen</p>			<b>Configuratie Management kent twee hoofdaandachtsgebieden</b> <ul style="list-style-type: none"><li>• Library Management (werkproducten kunnen terugvinden, ook eerdere versies; welke werkproducten horen bij elkaar)</li><li>• Change Control (beheerst omgaan met wijzigingen)</li></ul> <b>Literatuur</b> <ul style="list-style-type: none"><li>• Key Practices of the Capability Maturity Model v1.1, CMU/SEI-93-TR-25, <a href="http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf">www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf</a></li><li>• A Software Engineering Body of Knowledge v1.0, CMU/SEI-99-TR-004, <a href="http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr004.pdf">www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr004.pdf</a></li></ul> <b>Templates en overige middelen</b>		
<b>INPUT</b>		<b>ACTIE</b>		<b>OUTPUT</b>	
<b>Het inrichten van configuratie management</b>					
<ul style="list-style-type: none"><li>• De eisen die de eigen situatie stelt</li></ul>		<ul style="list-style-type: none"><li>• Documenteer de manier waarop met Configuratie Management wordt omgegaan (strategie)</li><li>• Richt een library systeem in</li><li>• Definieer de Configuration Items die er tijdens de ontwikkeling zijn. <i>CI = Configuration Item, is elke losse publicatie, procedure, beschrijving, programmadeel, sourcecode die vaak gezamenlijk maar ook apart verspreid zou kunnen worden</i></li><li>• Documenteer een procedure voor wijzigingen en probleemrapportages</li><li>• Documenteer een release procedure</li></ul>		<ul style="list-style-type: none"><li>• Library systeem (met onderscheid tussen ontwikkel- en exploitatieomgeving)</li><li>• Overzicht met CI's</li><li>• Wijzigingsprocedure</li><li>• Release procedure</li></ul>	
<b>Het uitvoeren van configuratie management</b>					
<ul style="list-style-type: none"><li>• Configuration items</li><li>• Wijzigingsverzoeken, (CR = Change Request)</li><li>• Probleemmeldingen, (PR = Problem Report)</li></ul>		<ul style="list-style-type: none"><li>• Beheer de library met configuration items</li><li>• Gebruik procedures voor wijzigingen en probleemrapportages</li><li>• Bewaak de status / voortgang van de PR's en CR's</li></ul>		<ul style="list-style-type: none"><li>• Status overzicht van de library</li><li>• Overzichten van CR's en PR's</li></ul>	
Stichting SPIDER		Werkgroep "SPI in Kleine Organisaties"		Copyright 2002	

**AANBEVELINGEN****Inrichten van Library Management**

- Start het beheer met een eenvoudige directory structuur en probeer hier de aanpak uit.  
Kies pas daarna voor een Configuratie Management-tool (je kunt dan tool-eisen beter definiëren)
- Creëer twee domeinen:
  - één voor de exploitatie van het product (bijv. gebruik, verkoop) en één voor de ontwikkeling (bevat zowel project- als productgegevens). Vóór uitvoering van een ontwikkelingsproject worden de benodigde productgegevens gekopieerd naar het ontwikkeldomein; ontwikkelaars voeren hier de wijzigingen door
  - na acceptatie van de wijzigingen wordt de nieuwe versie weer overgeheveld naar het exploitatie-domein
- Documenteer per CI hoe deze 'goedgekeurd' wordt (bijvoorbeeld na review, inspectie, testen, etc.)
- Documenteer hoe producten 'gereleased' worden: bouw het product in ieder geval op vanuit het library systeem
- Hanteer twee statussen Configuration Items (CI's): 'in ontwikkeling' en 'goedgekeurd'

**Inrichten Change Control**

- Documenteer hoe met wijzigingen (Change Requests of CR's) en probleem rapportages (Problem Reports of PR's) omgegaan wordt
- Leg vast wie er meebeslissen over wijzigingen oftewel wie deel uitmaken van de Change Control Board

**Uitvoeren van Library Management**

- Maak iemand verantwoordelijk voor het beheer (library management), zowel administratieve kant (wat zijn de items en waar kun je ze vinden), als ook de fysieke plaats van opslag
- Breng alle potentieel veranderbare producten onder configuration management: documenten, sources, installables, ontwikkelomgeving (toolversies, hw/sw-platforms), "installatie bouwomgeving", uitleveromgeving, 'testware' (plannen, testdata, tools, testrapporten, etc.)  
Vergeet vooral de speciale files/tools niet, die nodig zijn om de "build" te kunnen reproduceren op een "onafhankelijke" machine  
Het kan nuttig zijn de "reconstructie" van de build op een onafhankelijke machine zo nu en dan eens te "bewijzen"
- Bij publicaties / releases is belangrijk dat men kan terugtraceren welke wijzigingen zijn doorgevoerd  
(In eerste instantie zal dit requirements, c.q. CR's en PR's betreffen, maar naarmate professioneler gewerkt wordt ook wijzigingen in ontwerp, files, testware, etc.)

**Overig**

- Naarmate de projecten/teams groter worden of niet meer in dezelfde ruimte zitten, is het noodzakelijk dat de Configuratie Management-afspraken aangescherpt worden.
- M.b.t. beheer van de werkomgeving:
  - verplaats goedgekeurde versies naar de archiefomgeving en verwijder oudere versies.
  - bewaar in de werkomgeving maximaal: de laatste operationele versie; de laatste geaccepteerde versie en de werkversie